AMENDMENTS TO THE CLAIMS

1.   (Currently Amended) A method in a computer system for implementing a circular buffer, comprising:

        providing a buffer of words having a pointer for pointing to a word within the buffer and having a size;

        storing in a number of forwarding words, located past an end of the buffer, pointers to ~~locations~~ words at the other end of the buffer, a maximum number of words of the buffer to be accessed at a time corresponding to the number of forwarding words;

        enabling forwarding in the pointers;

        accessing a number of words starting at the word pointed to by the pointer modulo the size of the buffer wherein when a word within the buffer is to be accessed, accessing that word directly and when a forwarding word is to be accessed, directing the access to the ~~pointed to location~~ word at the other end of the buffer pointed to by the pointer stored in the accessed forwarding word; and

        incrementing the pointer by the number of words being accessed

        so that the buffer can be accessed without checking for the end of the buffer.

2.   (Original) The method of claim 1 wherein the buffer is pointed to by a write pointer whose value modulo a size of the buffer indicates the starting position for storing data in the buffer.

3.   (Original) The method of claim 1 wherein the buffer is pointed to by a read pointer whose value modulo a size of the buffer indicates the starting position for reading data from the buffer.

4.   (Original) The method of claim 1 wherein the access is a read.

5.   (Original) The method of claim 1 wherein the access is a write.

6.   (Original) The method of claim 1 wherein the access is using a pointer.

7.   (Original) The method of claim 6 wherein the pointer is a write pointer.

8.   (Original) The method of claim 6 wherein the pointer is a read pointer.

9.   (Original) The   method   of   claim   6   wherein   the   pointer   has   a synchronization access mode.

10.   (Currently Amended) The method of claim 9 wherein the synchronization access mode is sync.

11.   (Currently Amended) The method of claim 9 wherein the synchronization access mode is normal.

12.   (Currently Amended) The method of claim 9 wherein the synchronization access mode can be set.

13.   (Original) The method of claim 1 wherein the access does not include code for detecting the end of the buffer.

14.   (Original) The method of claim 1 further comprising:
when adding data to the buffer,
          receiving an indication of data to be written, the data having a size;
          fetching a write pointer;
          adding an indication of the size of the data to the write pointer; and

copying the data into the buffer starting at a location indicated by the fetched write pointer.

15. (Original) The method of claim 14 wherein the fetching and adding includes executing a fetch and add operation.

16. (Original) The method of claim 14 wherein when the copying would occur in a word located past an end of the buffer, the copying automatically circles to the other end of the buffer.

17. (Original) The method of claim 14 wherein the adding includes calculating a modulo of a sum of the addition and a size of the buffer.

18. (Original) The method of claim 1 further comprising:
when reading data from the buffer,

    receiving an indication of a location where read data is to be stored;

    fetching a read pointer;

    reading a size of the data to be read from the buffer; and

    copying data from the buffer to the indicated location.

19. (Original) The method of claim 18 further comprising setting the read pointer to a sum of the read pointer and the size of the data modulo a size of the buffer.

20. (Original) The method of claim 18 wherein the read pointer is accessed with a synchronization access mode of sync.

21. (Original) The method of claim 18 wherein the data is read from the buffer using an access control mode of the read pointer.

22.   (Original) The method of claim 1 wherein when the access has a synchronization access mode of sync, read access to a location in the buffer is permitted only when the location is full.

23.   (Original) The method of claim 22 wherein after the read access, the location is set to empty.

24.   (Original) The method of claim 1 wherein when the access has a synchronization access mode of sync, write access to a location in the buffer is permitted only when the location is empty.

25.   (Original) The method of claim 24 wherein after the write access, the location is set to full.

26.   (Original) The method of claim 1 including storing a pointer to an invalid location in a location adjacent to the forwarding words with forwarding of that location enabled so that when the location adjacent to the forwarding words is accessed, an exception is raised.

27.   (Original) The method of claim 1 wherein the buffer is accessed by multiple readers and writers.

28.   (Original) The method of claim 1 wherein the buffer is accessed by multiple producers.

29.   (Original) The method of claim 1 wherein the buffer is accessed by multiple consumers.

30. – 35. (Canceled)

36.   (Currently Amended) A computer system for implementing a circular buffer, the computer system having a processor, the system comprising:

a component that stores in forwarding words located past an end of a buffer pointers to locations at the other end of the buffer and enables forwarding in the pointers;

a component that accesses the buffer using an access pointer;

a component that increments the access pointer by the number of words accessed so that the incremented pointer points to a location for the next access; and

a component that, when a forwarding word is accessed, directs the access to the pointed to location at the other end of the buffer.

37.   (Original) The system of claim 36 wherein the buffer is accessed by multiple readers and writers.

38.   (Original) The system of claim 36 wherein the buffer is pointed to by a read pointer whose value modulo a size of the buffer indicates the starting position for reading data from the buffer.

39.   (Original) The system of claim 36 wherein the access is a read.

40.   (Original) The system of claim 36 wherein the access is a write.

41.   (Original) The system of claim 36 wherein the access is using a pointer.

42.   (Original) The system of claim 41 wherein the pointer is a write pointer.

43.   (Original) The system of claim 41 wherein the pointer is a read pointer.

44.   (Original) The system of claim 41 wherein the pointer has a synchronization access mode.

45.   (Currently Amended) The system of claim 44 wherein the synchronization access mode is sync.

46.   (Currently Amended) The system of claim 44 wherein the synchronization access mode is normal.

47.   (Currently Amended) The system of claim 44 wherein the synchronization access mode can be set.

48.   (Original) The system of claim 36 wherein the access does not include code for detecting the end of the buffer.

49.   (Currently Amended) A computer-readable storage medium for implementing a circular buffer, comprising:
    a buffer with storage locations, the buffer having a beginning and an end and having an access pointer, the access pointer pointing to a word to be accessed such that when then the buffer is accessed, the access pointer is incremented by the number of words being accessed so that the access pointer points to a location for the next access;
    a forwarding word adjacent to the end of the buffer; and
    a pointer in the forwarding word pointing to the beginning of the buffer
    so that when the forwarding word is accessed, the access can be redirected to the beginning of the buffer.

50.   (Currently Amended) The computer-readable storage medium of claim 49 further comprising multiple forwarding words wherein each forwarding word has a pointer to a storage location.

51.   (Currently Amended) The computer-readable storage medium of claim 50 wherein each forwarding word has forwarding enabled.

52.   (Currently Amended) The computer-readable storage medium of claim 49 wherein each forwarding word has forwarding enabled.

53.   (Currently Amended) The computer-readable storage medium of claim 49 further comprisingwherein the access pointer is a read pointer.

54.   (Currently Amended) The computer-readable storage medium of claim 53 wherein the value of the read pointer modulo a size of the buffer indicates a starting position for reading data from the buffer.

55.   (Currently Amended) The computer-readable storage medium of claim 49 further comprisingwherein the access pointer is a write pointer.

56.   (Currently Amended) The computer-readable storage medium of claim 55 wherein the value of the write pointer modulo a size of the buffer indicates a starting position for storing data in the buffer.

57. – 62. (Canceled)